Numerical Optimisation: Least squares

Marta M. Betcke
m.betcke@ucl.ac.uk,

Kiko Rullan
f.rullan@cs.ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 10 & 11

Least squares problem

Least squares is a problem where the objective function has the following special form

$$f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x),$$

where each r_j is a smooth function from $\mathbb{R}^n \to \mathbb{R}$. We refer to each of the r_j as a *residual*, and we assume that $m \geq n$.

Least squares problems are ubiquitous in applications, where the discrepancy between the model and the observed behaviour is minimised.

Let's assemble the individual components r_j into the *residual vector* $r : \mathbb{R}^n \to \mathbb{R}^m$

$$r(x) = (r_1(x), r_2(x), \dots, r_m(x))^{\mathrm{T}}.$$

Using this vector, f becomes $f(x) = \frac{1}{2}\|r(x)\|_2^2$. The derivatives of f can be calculated with help of the Jacobian

$$J(x) = \left[\frac{\partial r_j}{\partial x_i}\right]_{ij} = \begin{bmatrix} \nabla r_1(x)^{\mathrm{T}} \\ \nabla r_2(x)^{\mathrm{T}} \\ \vdots \\ \nabla r_m(x)^{\mathrm{T}} \end{bmatrix}.$$

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x)\nabla r_j(x) = J(x)^{\mathrm{T}} r(x),$$

$$\nabla^2 f(x) = \sum_{j=1}^{m} \nabla r_j(x)\nabla r_j(x)^{\mathrm{T}} + \sum_{j=1}^{m} r_j(x)\nabla^2 r_j(x)$$

$$= J(x)^{\mathrm{T}} J(x) + \sum_{j=1}^{m} r_j(x)\nabla^2 r_j(x),$$

Example

Model of concentration of drug in bloodstream

$$\phi(x;t) = x_1 + tx_2 + t^2 x_3 + x_4 \exp^{-x_5 t}.$$

Find a set of parameters x so that the model best matches the data solving

$$\frac{1}{2}\sum_{j=1}^{m}(\underbrace{\phi(x,t_j) - y_j}_{=:r_j(x)})^2.$$

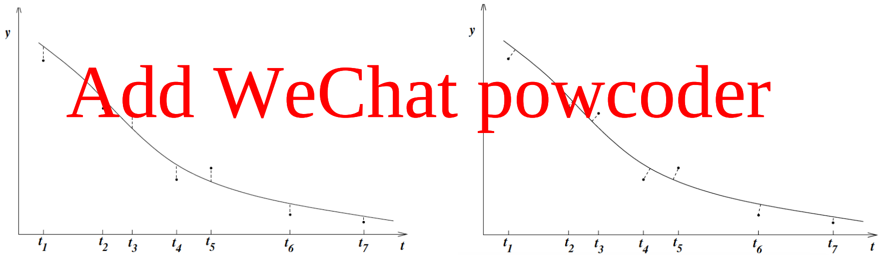

Figure: Nocedal Wright Fig 10.1 (left), Fig 10.2 (right)

Bayes' theorem

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} \propto \pi(y|x)\pi(x).$$

Denote the discrepancy between the model and the measurement at data point t_j as

$$\epsilon_j = \phi(x; t_j) - y_j.$$

Assume that ϵ_js are independent and identically distributed with variance σ^2 and probability density function $g_\sigma(\cdot)$. The *likelyhood* of a particular set of observations $y_j, j = 1, 2, \dots, m$ given the parameter vector x is given by

$$\pi(y|x) = \prod_{j=1}^{m} g_\sigma(\epsilon_j) = \prod_{j=1}^{m} g_\sigma(\phi(x; t_j) - y_j).$$

The *maximum a posteriori probability* (MAP) estimate vs the maximum likelyhood estimate

$$x_{\mathrm{MAP}} = \max_x \pi(x|y) = \max_x \pi(y|x)\pi(x).$$

If g_σ is a *normal* distribution

$$g_\sigma(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$$

and x is uniformly distributed i.e. $\pi(x) = const$, the MAP estimate reads

$$\begin{aligned} x_{\mathrm{MAP}} &= \max_x \frac{1}{(\sqrt{2\pi}\sigma)^m} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^m (\phi(x;t_j) - y_j)^2\right) \\ &= \min_x \sum_{j=1}^m (\phi(x;t_j) - y_j)^2. \end{aligned}$$

If $\phi(x;t)$ is linear function in x, the least squares problem becomes linear.

The residuals $r_j(x) = \phi(x;t_j) - y_j$ are linear and in the vectorized notation

$$r(x) = Jx - y,$$

where

- the vector of measurements $y = (y_1, y_2, \ldots, y_m)^{\mathrm{T}} = r(0)^{\mathrm{T}}$
- the matrix J with rows $J_{j,:}$: $\phi(x;t_j) = J_{j,:}x$

are both independent of x.

The linear least squares has the form

$$f(x) = \frac{1}{2}\|Jx - y\|^2.$$

The gradient and Hessian are

$$\nabla f(x) = J^{\mathrm{T}}(Jx - y), \quad \nabla^2 f(x) = J^{\mathrm{T}} J.$$

Note: $f(x)$ is convex i.e. the stationary point is the global minimiser $\nabla f(x^\star) = 0$.

Normal equations

$$\nabla f(x^\star) = J^{\mathrm{T}}(Jx^\star - y) = 0 \quad \Leftrightarrow \quad J^{\mathrm{T}} J x^\star = J^{\mathrm{T}} y.$$

Roadmap: solution of linear least squares

- Solve the normal equations $J^{\mathrm{T}} J x^\star = J^{\mathrm{T}} y$
  + If $m \gg n$, computing $J^{\mathrm{T}} J$ explicitly results is a smaller matrix easier to store than J. This can be solved by e.g. Cholesky decomposition.
  - Formulating $J^{\mathrm{T}} J$ squares the condition number.
  + For regularised *ill-posed* problems squaring of the condition number may not be an issue.
  - If J is rank deficient (or ill conditioned) Cholesky decomposition will require pivoting.
- Solve the least squares $x^\star = \arg\min_x \|Jx - b\|^2$
  + If J is of moderate size you can use direct methods like QR decomposition or SVD decomposition.
  + If J is large and sparse or given in operator form i.e. $x \to Jx$ use iterative methods like CGLS or LSQR.
  + Does not square the condition number.
  + In particular SVD and iterative methods e.g. LSQR can easily deal with ill-conditioning.

Let

$$JP = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = [\ \underbrace{Q_1}_{\in\mathbb{R}^{m\times n}}\quad \underbrace{Q_2}_{\in\mathbb{R}^{m\times(m-n)}}\ ]\begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R,$$

where

- $P \in \mathbb{R}^{n\times n}$ column permutation matrix (hence orthogonal),
- $Q \in \mathbb{R}^{m\times m}$ orthogonal matrix,
- $R \in \mathbb{R}^{n\times n}$ upper triangular with positive diagonal.

Recall: Multiplication with orthogonal matrix preserves $\|\cdot\|_2$.

$$\begin{aligned} \|Jx - y\|_2^2 &= \|Q^{\mathrm{T}}(J\overbrace{PP^{\mathrm{T}}}^{=I}x - y)\|_2^2 = \|(\overbrace{Q^{\mathrm{T}}JP}^{=[R\ 0]^{\mathrm{T}}})P^{\mathrm{T}}x - Q^{\mathrm{T}}y\|_2^2 \\ &= \|RP^{\mathrm{T}}x - Q_1^{\mathrm{T}}y\|_2^2 + \|Q_2^{\mathrm{T}}y\|_2^2. \end{aligned}$$

Solution: $x^\star = PR^{-1}Q_1^{\mathrm{T}}y$. In practice we perform backsubstitution on $Rz = Q_1^{\mathrm{T}}y$ and permute for $x^\star = Pz$.

Let

$$J = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^{\mathrm{T}} = [\, \underbrace{U_1}_{\in \mathbb{R}^{m\times n}} \quad \underbrace{U_2}_{\in \mathbb{R}^{m\times (m-n)}} \,] \begin{bmatrix} S \\ 0 \end{bmatrix} V^{\mathrm{T}} = U_1 S V^{\mathrm{T}},$$

where

- $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices,
- $S \in \mathbb{R}^{n\times n}$ diagonal matrix with elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$.

$$\begin{aligned} \|Jx - y\|_2^2 &= \|U^{\mathrm{T}}(J \overbrace{VV^{\mathrm{T}}}^{=I} x - y)\|_2^2 = \|\overbrace{(U^{\mathrm{T}} J V)}^{=[S,0]^{\mathrm{T}}} V^{\mathrm{T}} x - U^{\mathrm{T}} y\|_2^2 \\ &= \|SV^{\mathrm{T}} x - U_1^{\mathrm{T}} y\|_2^2 + \|U_2^{\mathrm{T}} y\|_2^2. \end{aligned}$$

Solution: $x^\star = VS^{-1}U_1^{\mathrm{T}} y = \sum_{i=1}^{n} \frac{u_i^{\mathrm{T}} y}{\sigma_i} v_i$. If σ_i are small, they would undue amplify the noise and can be omitted from the sum. Picard condition: $|u_i^{\mathrm{T}} y|$ should decay faster than σ_i.

LSQR [Paige, Saunders '82]

LSQR applied to

$$\min_f \|Af - g\|_2^2 + \tau \|f\|_2^2 = \min_f \left\| \begin{bmatrix} A \\ \sqrt{\tau} I \end{bmatrix} f - \begin{bmatrix} g \\ 0 \end{bmatrix} \right\|_2^2,$$

where τ is an optional damping parameter, is analytically equivalent to CG applied to the normal equations. However, it avoids forming them (hence no condition number squaring) using the Golub–Kahan bidiagonalization [Golub,Kahan '65].

G-K bidiagonalisation yields the projected least squares problem

$$\min_{y_i} \left\| \begin{bmatrix} B_i \\ \sqrt{\tau} I \end{bmatrix} y_i - \beta_1 e_1 \right\|, \qquad \text{(P-LS)}$$

which is then solved using QR decomposition yielding the approximation for the solution of the original problem, $f_i = V_i y_i$.

Golub-Kahan bidiagonalization

G-K bidiagonalization with a starting vector g for $\min_f \|g - Af\|$

$$
\begin{aligned}
U_{i+1}(\beta_1 e_1) &= g \\
AV_i &= U_{i+1} B_i \\
A^{\mathrm{T}} U_{i+1} &= V_i B_i^{\mathrm{T}} + \alpha_{i+1} v_{i+1} e_{i+1}^{\mathrm{T}},
\end{aligned}
$$

e_i: ith canonical basis vector, $\alpha_i \geq 0$ and $\beta_i \geq 0$ chosen such that $\|u_i\| = \|v_i\| = 1$ and $U_i^{\mathrm{T}} U = I$, $V_i^{\mathrm{T}} V = I$,

$$
B_i = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_i \\ & & & \beta_{i+1} \end{bmatrix}, \quad U_i = [u_1, u_2, \ldots, u_i], \quad V_i = [v_1, v_2, \ldots, v_i]. \tag{1}
$$

Preconditioned LSQR

$$\hat{f} = \operatorname{argmin} \|g - AL^{-1}\hat{f}\|.$$

The corresponding normal equation is exactly the split preconditioned normal equation

$$L^{-\mathrm{T}}A^{\mathrm{T}}AL^{-1}\hat{f} = L^{-\mathrm{T}}A^{\mathrm{T}}g, \qquad f = L^{-1}\hat{f}. \tag{2}$$

Similarly as for CG, the LSQR algorithm can be formulated without the need to provide a factorization $L^{\mathrm{T}}L$ of M, the MLSQR algorithm [Arridge, B, Harhanen '14].

Initialization:
$\beta_1 u_1 = g$
$\tilde{p} = A^{\mathrm{T}} u_1$
$\tilde{v}_1 = M^{-1}\tilde{p}$, $\alpha_1 = (\tilde{v}_1, \tilde{p})^{1/2}$, $\tilde{v}_1 = \tilde{v}_1/\alpha_1$
$\tilde{w}_1 = \tilde{v}_1,\ f_0 = 0,\ \bar{\phi}_1 = \beta_1,\ \bar{\rho}_1 = \alpha_1$
for $i = 1, 2, \ldots$ **do**
  Bidiagonalization:
  $\beta_{i+1} u_{i+1} = A\tilde{v}_i - \alpha_i u_i$
  $\tilde{p} = A^{\mathrm{T}} u_{i+1} - \beta_{i+1}\tilde{p}$
  $\tilde{v}_{i+1} = M^{-1}\tilde{p},\ \alpha_{i+1} = (\tilde{v}_{i+1}, \tilde{p})^{1/2},$
  $\tilde{p} = \tilde{p}/\alpha_{i+1},\ \tilde{v}_{i+1} = \tilde{v}_{i+1}/\alpha_{i+1}$
  Orthogonal transformation: smart QR, see [Paige Sanders '82]
  Update:
  $f_i = f_{i-1} + (\phi_i/\rho_i)\tilde{w}_i$
  $\tilde{w}_{i+1} = \tilde{v}_{i+1} - (\theta_{i+1}/\rho_i)\tilde{w}_i$
  Break if stopping criterion satisfied
end for

LSQR with explicit regularization ($\tau \neq 0$)

- In preconditioned formulation, Tikhonov (explicit) regularization amounts to damping. For a fixed value of τ, damping can be easily incorporated in LSQR at the cost of doubling the number of Givens rotations [Paige, Saunders '82].
- Due to the shift invariance of Krylov spaces, V_i are the same for any τ. If V_i are stored (expensive for many/long vectors), the projected least squares problem (P-LS) can be efficiently solved for multiple values of τ.
- Solving (P-LS) with a variable τ is discussed in [Bjorck '88] using singular value decomposition of the bidiagonal matrix B_i (no efficient SVD update even though B_i simply expands by a row and a column in each iteration). Those quantities can be obtained at the cost $\mathcal{O}(i^2)$ at the i^{th} iteration.
- For larger i, the algorithm described in [Elden '77] for the least squares solution of (P-LS) at the cost of $\mathcal{O}(i)$ for each value of τ is the preferable option.

Stopping LSQR / MLSQR

[Saunders Paige '82] discusses three stopping criteria:

S1: $\|\bar{r}_i\| \leq \mathrm{BTOL}\|g\| + \mathrm{ATOL}\|\bar{A}\|\|f_i\|$ (consistent systems),

S2: $\frac{\|\bar{A}^T \bar{r}_i\|}{\|\bar{A}\|\|\bar{r}_i\|} \leq \mathrm{ATOL}$ (inconsistent systems),

S3: $\mathrm{cond}(\bar{A}) \geq \mathrm{CONLIM}$ (both).

where $\bar{r}_i := \bar{g} - \bar{A}f_i$ with $\bar{A} = \begin{bmatrix} A \\ \sqrt{\tau}L \end{bmatrix}, \quad \bar{g} = \begin{bmatrix} g \\ 0 \end{bmatrix}$.

[Arridge, B, Harhanen '14] uses Morozov discrepancy principle (suitable for ill-posed problems)

S4: $\|r_i\| \leq \eta\delta, \quad \eta > 1$,

where $r_i := g - Af_i$, δ is the (estimated) noise level, $\eta > 1$ prevents overregularization

+ if $\tau = 0$, $r_i = \bar{r}_i$ and the sequence $\|r_i\| = \|\bar{r}_i\|$ is monotonically decreasing. Moreover if initialised with f_0, $\|f_i\|$ is strictly monotonically growing (relevant for damped problem),
+ priorconditioning does not alter the residual.

Gauss-Newton (GN) can be viewed as a modified Newton method with line search.

Recall the specific form of gradient and Hessian of least squares problems

$$\nabla f(x) = J(x)^{\mathrm{T}} r(x), \quad \nabla^2 f(x) = J(x)^{\mathrm{T}} J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x).$$

Substituting into the Newton equation

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

and using the approximation $\nabla^2 f(x_k) \approx J(x_k)^{\mathrm{T}} J(x_k)$ we obtain

$$\underbrace{J(x_k)^{\mathrm{T}}}_{=:J_k^{\mathrm{T}}} J(x_k) p_k^{\mathrm{GN}} = -J(x_k)^{\mathrm{T}} \underbrace{r(x_k)}_{=:r_k}.$$

Implementations of GN usually perform a line search along p_k^{GN} requiring the step length to satisfy e.g. Armijo or Wolfe conditions.

- Does not require computation of the individual Hessians $\nabla^2 r_j, j = 1, \dots, m$. If the Jacobian J_k has been computed when evaluating the gradient no other derivatives are needed.

- Frequently the first term $J_k^{\mathrm{T}} J_k$ dominates the second term in the Hessian i.e. $|r_j(x)|\|\nabla^2 r_j(x)\|$ are much smaller than the eigenvalues $J^{\mathrm{T}} J$. This happens if either the residual r_j or $\|\nabla^2 r_j\|$ are small. Thus $J_k^{\mathrm{T}} J_k$ is a good approximation to $\nabla^2 f_k$ and the convergence rate of GN is close to Newton.

- If J_k has full rank and $\nabla f_k \neq 0$, p_k^{GN} is a descent direction for f and hence suitable for line search

$$\begin{aligned}(p_k^{\mathrm{GN}})^{\mathrm{T}} \nabla f_k &= (p_k^{\mathrm{GN}})^{\mathrm{T}} J_k^{\mathrm{T}} r_k = -(p_k^{\mathrm{GN}})^{\mathrm{T}} J_k^{\mathrm{T}} J_k p_k^{\mathrm{GN}} \\ &= -\|J_k p_k^{\mathrm{GN}}\|^2 \le 0.\end{aligned}$$

  The final inequality is strict unless $J_k p_k^{\mathrm{GN}} = 0$ in which case by the GN equation and J_k being full-rank we have $0 = J_k^{\mathrm{T}} r_k = \nabla f_k$ and x_k is a stationary point.

Interpretation of GN step

The GN equation

$$J_k^{\mathrm{T}} J_k p_k^{\mathrm{GN}} = -J_k^{\mathrm{T}} r_k$$

is exactly the normal equation for the linear least squares problem

$$\min_p \frac{1}{2}\|J_k p + r_k\|^2.$$

Hence, we can find the GN direction p_k^{GN} solving this linear least squares problem using any of the techniques discussed before.

We can view GN equation as obtained from a linear model for the vector function $r(x_k + p) \approx r_k + J_k p$,

$$f(x_k + p) = \frac{1}{2}\|r_k(x_k + p)\|^2 \approx \frac{1}{2}\|J_k p + r_k\|^2$$

and $p_k^{\mathrm{GN}} = \arg\min_p \frac{1}{2}\|J_k p + r_k\|^2$.

The global convergence is a consequence of the convergence theorem for line search methods [Zoutendijk].

To satisfy the conditions of Zoutendijk's theorem, we need to make following assumptions:

- r_j is Lipschitz continuously differentiable in a neighbourhood $\mathcal{M}$ of the bounded level set $\mathcal{L} = \{x : f(x) \le f(x_0)\}$

- $J(x)$ satisfies the uniform full-rank condition, $\gamma > 0$

$$\|J(x)z\| \ge \gamma \|z\|, \quad \forall x \in \mathcal{M}.$$

Then for the iterates x_k generated by the GN method with step length satisfying Wolfe conditions, we have

$$\lim_{k\to\infty} J_k^{\mathrm{T}} r_k = \nabla f(x_k) = 0.$$

Similarly as for the line search, we check that the angle $\theta_k = \angle(p_k^{\mathrm{GN}}, -\nabla f_k)$ is uniformly bounded away from $\pi/2$

$$\cos\theta_k = -\frac{(p_k^{\mathrm{GN}})^{\mathrm{T}}\nabla f_k}{\|p_k^{\mathrm{GN}}\|\|\nabla f_k\|} = \frac{\|J_k p_k^{\mathrm{GN}}\|^2}{\|p_k^{\mathrm{GN}}\|\|J_k^{\mathrm{T}} J_k p_k^{\mathrm{GN}}\|} \geq \frac{\gamma^2\|p_k^{\mathrm{GN}}\|^2}{\beta^2\|p_k^{\mathrm{GN}}\|^2} = \frac{\gamma^2}{\beta^2} > 0,$$

where $\|J_k(x)\| \leq \beta < \infty, \ \forall x \in \mathcal{L}$ is a consequence of boundedness of the level set $\mathcal{L}$ and Lipschitz continuous differentiability of $r_j, j = 1, \dots, m$.

Then from $\sum_{k\geq 0} \cos^2\theta_k \|\nabla f_k\|^2 < \infty$ in Zoutendijk's theorem it follows $\nabla f_k \to 0$.

If J_k for some k is rank deficient, the matrix $J_k^{\mathrm{T}} J_k$ in GN equation is singular and the system has infinitely many solutions, however $\cos\theta_k$ is not necessarily bounded away from 0.

The convergence of GN can be rapid if $J_k^{\mathrm{T}} J_k$ dominates the second term in the Hessian. Similarly as showing the convergence rate of Newton iteration, if x_k is sufficiently close to $x^\star$, $J(x)$ satisfies the uniform full-rank condition, we have for a unit step in GN direction

$$x_k + p_k^{\mathrm{GN}} - x^\star = x_k - x^\star - [\overbrace{J^{\mathrm{T}} J(x_k)}^{:=J(x_k)^{\mathrm{T}} J(x_k)}]^{-1} \nabla f(x_k)$$

$$= [J^{\mathrm{T}} J(x_k)]^{-1} \left[J^{\mathrm{T}} J(x_k)(x_k - x^\star) + \underbrace{\nabla f(x^\star)}_{=0} - \nabla f(x_k) \right].$$

Using $H(x)$ to denote the second term in the Hessian, it follows from Taylor theorem that

$$\nabla f(x_k) - \nabla f(x^\star) = \int_0^1 J^{\mathrm{T}} J(x^\star + t(x_k - x^\star))(x_k - x^\star) dt + \int_0^1 H(x^\star + t(x_k - x^\star))(x_k - x^\star) dt,$$

Putting everything together and assuming Lipschitz continuity of $H(\cdot)$ near $x^\star$ and using L.c.d. of $r_j \Rightarrow$ L.c. of $J^{\mathrm{T}} r(x)$

$$\|x_k + p_k^{\mathrm{GN}} - x^\star\| \leq \int_0^1 \|[J^{\mathrm{T}} J(x_k)]^{-1} H(x^\star + t(x_k - x^\star))\| \|x_k - x^\star\| dt$$

$$+ \int_0^1 \| \underbrace{[J^{\mathrm{T}} J(x_k)]^{-1}}_{\|\cdot\| \leq \gamma^{-2}} \underbrace{\left(J^{\mathrm{T}} J(x^\star + t(x_k - x^\star)) - J^{\mathrm{T}} J(x_k)\right)}_{\|\cdot\| = \mathcal{O}(\|x_k - x^\star\|) \text{ would need L.c. of } J^{\mathrm{T}} J(x)} \| \|x_k - x^\star\| dt$$

$$\approx \|[J^{\mathrm{T}} J(x^\star)]^{-1} H(x^\star)\| \|x_k - x^\star\| + \mathcal{O}(\|x_k - x^\star\|^2).$$

Hence, if $\|[J^{\mathrm{T}} J(x^\star)]^{-1} H(x^\star)\| \ll 1$, we can expect GN to converge quickly towards the solution $x^\star$. When $H(x^\star) = 0$, the convergence is quadratic (Newton).

When the Jacobian $J(x)$ is large and sparse, the exact solve of GN equation can be replaced by an *inexact* solve as in inexact Newton methods but with the true Hessian $\nabla^2 f(x_k)$ replaced with $J(x_k)^{\mathrm{T}} J(x_k)$. The positive semidefinitness of $J(x_k)^{\mathrm{T}} J(x_k)$ simplifies the algorithms. Instead of (preconditioned) CG, (preconditioned) LSQR should be used.

Levenberg-Marquardt (LM) method

Levenberg-Marquardt (LM) makes use of the same Hessian approximation as GN but within the framework of trust region methods. Trust region methods can cope with (nearly) rank-deficient Hessian, which is a weakness of GN.

The constraint model problem to be solved at each iteration

$$\min_p \frac{1}{2}\|J_k p + r_k\|^2, \quad \text{subject to } \|p\| \leq \Delta_k, \qquad \text{(CM-LM)}$$

where $\Delta_k > 0$ is the trust region radius.

Note: The least squares term corresponds to quadratic model

$$m_k(p) = \frac{1}{2}\|r_k\|^2 + p^{\mathrm{T}} J_k^{\mathrm{T}} r_k + \frac{1}{2} p^{\mathrm{T}} J_k^{\mathrm{T}} J_k p.$$

Solution of the constraint model problem

The solution of the constraint model problem (CM-LM) is an immediate consequence of the general result for trust region methods [More, Sorensen]:

- If the solution p_k^{GN} of the GN equation lies strictly inside the trust region i.e. $\|p_k^{\mathrm{GN}}\| < \Delta_k$, then $p_k^{\mathrm{LM}} = p_k^{\mathrm{GN}}$ solves (CM-LM).

- Otherwise, there is a $\lambda > 0$ such that the solution p_k^{LM} of (CM-LM) satisfies $\|p_k^{\mathrm{LM}}\| = \Delta_k$ and

$$(J_k^{\mathrm{T}} J_k + \lambda I) p_k^{\mathrm{LM}} = -J_k^{\mathrm{T}} r_k.$$

Note: The last equation is the normal equation to the linear least squares problem

$$\min_p \frac{1}{2} \left\| \begin{bmatrix} J_k \\ \sqrt{\lambda} I \end{bmatrix} p + \begin{bmatrix} r \\ 0 \end{bmatrix} \right\|^2,$$

which gives us a way of solving (CM-LM) without computing $J_k^{\mathrm{T}} J_k$.

Global convergence LM

Global convergence is a consequence of the corresponding trust region global convergence theorem.

To satisfy the conditions of that theorem we make the following assumptions:

- $\eta \in (0, \frac{1}{4})$ (for strong convergence)
- r_j is Lipschitz continuously differentiable in a neighbourhood $\mathcal{N}$ of the bounded level set $\mathcal{L} = \{x : f(x) \le f(x_0)\}$

- The approximate solution p_k of (CM-LM) satisfies

$$m_k(0) - m_k(p_k) \ge c_1 \|J_k^{\mathrm{T}} r_k\| \min\left(\Delta_k, \frac{\|J_k^{\mathrm{T}} r_k\|}{\|J_k^{\mathrm{T}} J_k\|}\right),$$

  for some constant $c_1 > 0$, and in addition $\|p_k\| \le \gamma \Delta_k$ for some constant $\gamma \ge 1$.

We then have that

$$\lim_{k \to \infty} \nabla f_k = \lim_{k \to \infty} J_k^{\mathrm{T}} r_k = 0.$$

- As for trust region methods, there is no need to evaluate the right hand side of the decrease condition, but it is sufficient to ensure reduction of at least the Cauchy point, which can be calculated inexpensively. If the iterative CG-Steighaus approach is used, this is guaranteed.

- The local convergence of LM is similar to GN. Near the solution $x^\star$, at which the first term $J(x^\star)^{\mathrm{T}} J(x^\star)$ of the Hessian $\nabla^2 f(x^\star)$ dominates the second term, the trust region becomes inactive and the algorithm takes GN steps giving rapid local convergence.

- [Paige, Saunders '82] Link to website with papers and codes `https://web.stanford.edu/group/SOL/software/lsqr/`
- [Bjorck '96] A. Bjorck, "Numerical Methods for Least Squares Problems", SIAM, 1996
- [Golub, Kahan '65] G.H. Golub and W. Kahan, "Calculating the singular values and pseudoinverse of a matrix", *SIAM J.Numer.Anal. 2 ,1965*
- [Elden '77] L. Elden, "Algorithms for the regularization of ill-conditioned least squares problems", *BIT, 17, 1977*
- [Arridge, B, Harhanen '14] S R Arridge, M M Betcke and L Harhanen, "Iterated preconditioned LSQR method for inverse problems on unstructured grids", *Inverse Problems*, 30(7) 2014
- [Hansen '98] P. C. Hansen "Rank-Deficient and Discrete Ill-Posed Problems", SIAM, 1998